# SUPPORT VECTOR MACHINES

DR H.K. LAM

Department of Informatics

King's College London

Office S1.15, Strand Campus
Email: hak-keung.lam@kcl.ac.uk

Pattern Recognition (6CCS3PRE/7CCSMPNN)

# Outline

1. Introduction

2. Linear SVMs: Linearly Separable Case

3. Linear SVMs: Non-separable Case

4. Nonlinear SVMs

5. Multi-class SVMs

6. Conclusion

# Introduction

**Support Vector Machines (SVMs):**

- Works in a similar concept of linear machines with margins.
- Relies on preprocessing the data in a high dimension using *Kernel* functions.
- Classifies two classes, i.e., binary classifier.
- Computes the optimal weights instead of through training.

# Introduction

- Support Vector Machines (SVMs) based on *Linear Discriminant Functions*
    - Include margins to optimise solution
    - Can allow errors to occurs in a controlled way
- Comparable to Neural Networks
    - Allow non-linear mappings in higher dimensional feature space through use of Kernel functions
    - Advantages over NNs as simpler to select models and less susceptible to over-fitting

# Introduction

**Two-Class Classification Problem:**

- Labelled training samples: $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$
  - $\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{bmatrix}$,
  - $y_i \in \{-1, 1\}$, $i$ = 1, 2, ..., $N$,
  - $N$ denotes the number of training samples.
- (Goal) Design a hyperplane $f(\mathbf{x}) = 0$ which can classify correctly (all) the training samples.

# Introduction

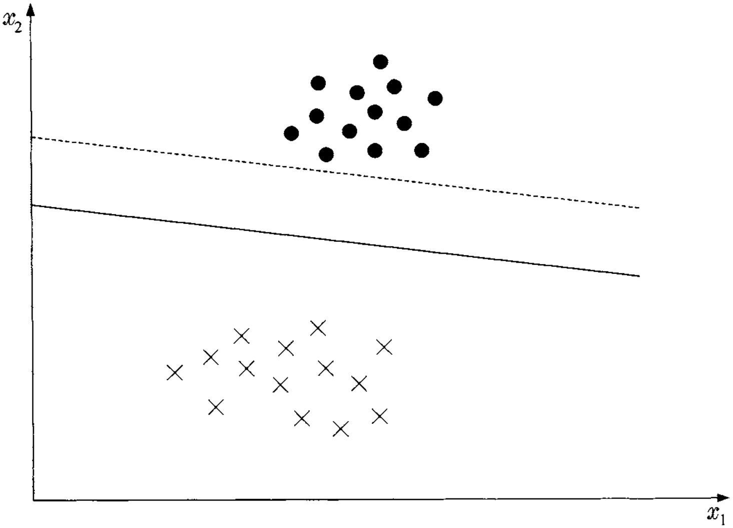


Figure 1: A diagram showing linearly separable two classes.

# Introduction

- How to design the optimal classifier, i.e., design the optimal hyperplane $f(\mathbf{x})$?
- Only optimal if
  - No errors, i.e., no mis-classification
  - Distance or margin between nearest *support vectors* and separating plane is maximal.
  - What is a support vector?
- Can be achieved graphically on a small data set.

# Introduction

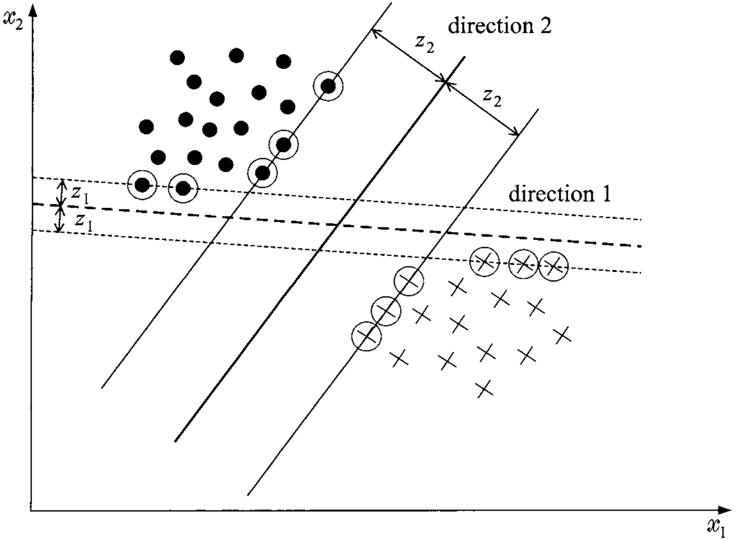


Figure 2: A diagram showing two linear classifiers with two margins.

# Linear SVMs

- Deals with linearly separable 2-class classification problem.
- Hyperplane: $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 = 0$ where $\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$.
- Find $\mathbf{w}$ and $w_0$ such that
  - the margin is optimal
  - $\begin{cases} \mathbf{w}^T\mathbf{x} + w_0 \geq 1, & \forall \mathbf{x} \in \text{class 1 } (\text{``}+1\text{''}) \\ \mathbf{w}^T\mathbf{x} + w_0 \leq -1, & \forall \mathbf{x} \in \text{class 2 } (\text{``}-1\text{''}) \end{cases}$

# Linear SVMs: Linearly Separable Case

**Optimal margin:**

- Distance of a point from a hyperplane: $z = \frac{|f(\mathbf{x})|}{\|\mathbf{w}\|}$.
- $\|\cdot\|$ is the $l^2$ norm operator (also known as Euclidean norm).
- Achieve a maximum margin (distance): find the largest margin $z$ between the hyperplane and support vectors.
- The margin is given by:

$$\begin{aligned}
&\min_{\mathbf{x}_i : y_i = -1} \frac{|f(\mathbf{x}_i)|}{\|\mathbf{w}\|} + \min_{\mathbf{x}_i : y_i = +1} \frac{|f(\mathbf{x}_i)|}{\|\mathbf{w}\|} \\
&= \frac{1}{\|\mathbf{w}\|}\left( \min_{\mathbf{x}_i : y_i = -1} |f(\mathbf{x}_i)| + \min_{\mathbf{x}_i : y_i = +1} |f(\mathbf{x}_i)| \right) \\
&= \frac{2}{\|\mathbf{w}\|}
\end{aligned}$$

# Linear SVMs: Linearly Separable Case

**Constrained optimisation problem:**

$$\boxed{\begin{array}{c} \min_{\mathbf{w},w_0} \quad J(\mathbf{w},w_0) = \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + w_0) \geq 1, \quad i = 1,2,\ldots,N \end{array}}$$

**Method of Lagrange multipliers:**

$$\text{Primal problem: } \mathscr{L}(\mathbf{w},w_0,\lambda) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \lambda_i(y_i(\mathbf{w}^T\mathbf{x}_i + w_0) - 1)$$

where $\lambda = \left[\begin{array}{cccc} \lambda_1 & \lambda_2 & \ldots & \lambda_N \end{array}\right]$.

The above primal problem can be transformed to the following dual problem:

$$\text{Dual problem: } \min_{\mathbf{w},w_0} \max_{\lambda \geq 0} \left(\frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \lambda_i(y_i(\mathbf{w}^T\mathbf{x}_i + w_0) - 1)\right)$$

# Linear SVMs: Linearly Separable Case

$$\frac{\partial \mathscr{L}(\mathbf{w}, w_0, \lambda)}{\partial \mathbf{w}} = \mathbf{0} \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{N} \lambda_i y_i \mathbf{x}_i \tag{1}$$

$$\frac{\partial \mathscr{L}(\mathbf{w}, w_0, \lambda)}{\partial w_0} = 0 \quad \Rightarrow \quad \sum_{i=1}^{N} \lambda_i y_i = 0 \tag{2}$$

Putting (1) and (2) into $\mathscr{L}(\mathbf{w}, w_0, \lambda)$, we have

$$\begin{aligned}
\mathscr{L}(\lambda) &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \lambda_i (y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1) \\
&= \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^{N} \lambda_i y_i \mathbf{w}^T \mathbf{x}_i - \sum_{i=1}^{N} \lambda_i y_i w_0 + \sum_{i=1}^{N} \lambda_i \\
&= \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^{N} \lambda_i y_i w_0 + \sum_{i=1}^{N} \lambda_i \\
&= \sum^{N} \lambda_i - \frac{1}{2} \sum^{N} \sum_{=1}^{N} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j .
\end{aligned}$$

# Linear SVMs: Linearly Separable Case

The dual problem is reduced to:

$$\max_{\lambda \geq 0} \left( \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \right) \tag{3}$$

$$\text{subject to } \sum_{i=1}^{N} \lambda_i y_i = 0,$$

$$\lambda_i \geq 0, i = 1, 2, \ldots, N$$

# Linear SVMs: Linearly Separable Case

- Solution $\lambda_i$ to (3) can be found by using *quadratic programming solver*

- It is a scalar function which does not depend explicitly on the dimension of the input space.

- The solution of the Lagrange multipliers $\lambda_i$ may not be unique but the hyperplane characterised by $\mathbf{w}$ and $w_0$ is unique.

- For those $\mathbf{x}_i$ with $\lambda_i \neq 0$, they are known as *support vectors*. As a result, $\mathbf{w} = \sum_{i=1}^{N_s} \lambda_i y_i \mathbf{x}_i$, $N_s \leq N$ denotes the number of support vectors. (Note: $\mathbf{x}_i$ here refers to a support vector not any $\mathbf{x}_i$ in the training samples)

- The support vectors lie on the two hyperplanes satisfying $\mathbf{w}^T\mathbf{x} + w_0 = \pm 1$ where $\mathbf{x} \in$ support vectors. (Why?)

# Linear SVMs: Linearly Separable Case

Rearranging the terms in $\mathscr{L}(\lambda)$,

$$\begin{aligned}\mathscr{L}(\lambda) &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N}\lambda_i(y_i(\mathbf{w}^T\mathbf{x}_i + w_0) - 1)\\ &= \frac{1}{2}\mathbf{w}^T\sum_{i=1}^{N}\lambda_i y_i \mathbf{x}_i + \frac{1}{2}\sum_{i=1}^{N}\lambda_i y_i w_0 - \sum_{i=1}^{N}\lambda_i(y_i(\mathbf{w}^T\mathbf{x}_i + w_0) - 1)\\ &= \frac{1}{2}\sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i=1}^{N}\lambda_i(y_i(\mathbf{w}^T\mathbf{x}_i + w_0) - 1)\end{aligned}$$

- To maximise $\mathscr{L}(\lambda)$ in $\lambda_i$, since $\lambda_i \geq 0$ and $y_i(\mathbf{w}^T\mathbf{x}_i + w_0) - 1 \geq 0$, one possibility is to have $\lambda_i(y_i(\mathbf{w}^T\mathbf{x}_i + w_0) - 1) = 0$ for all $y_i(\mathbf{x}_i + w_0) - 1 \geq 0$, $i$ = 1, 2, ..., $N$.

- This is supported by Karush-Kuhn-Tucker (KKT) conditions.

- As $y_i(\mathbf{w}^T\mathbf{x}_i + w_0) - 1 \geq 0$, it suggests that some $\lambda_i = 0$ for those $\mathbf{x}_i$ not being a support vector.

# Linear SVMs: Linearly Separable Case

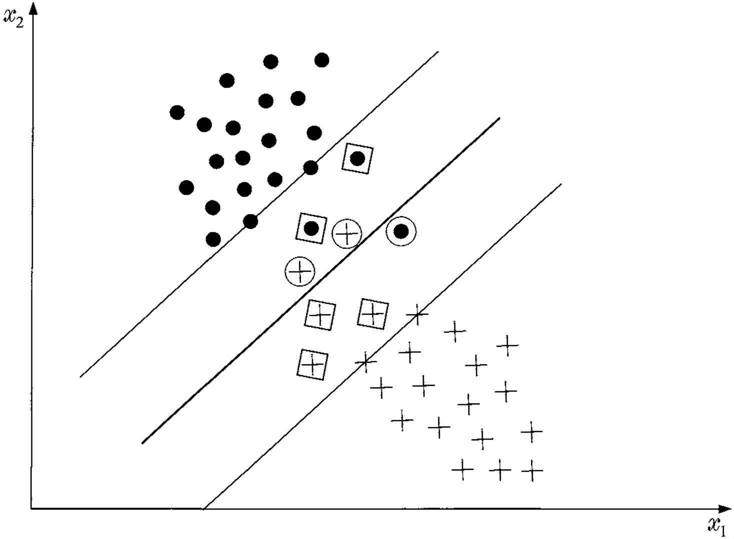
**Summary:** The linear SVM classifier (linearly separable case) can be found by solving the solution ($\mathbf{w}$, $w_0$ and $\lambda_i$) to the following conditions:

$$\boxed{\begin{aligned}\frac{\partial \mathscr{L}(\mathbf{w}, w_0, \lambda)}{\partial \mathbf{w}} &= \mathbf{0} \\ \frac{\partial \mathscr{L}(\mathbf{w}, w_0, \lambda)}{\partial w_0} &= 0 \\ \lambda_i &\geq 0, \quad i = 1, 2, \ldots, N \\ \lambda_i \big(y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1\big) &= 0, \quad i = 1, 2, \ldots, N\end{aligned}}$$

Hard classifier: $f(\mathbf{x}) = \text{sgn}(\mathbf{w}^T \mathbf{x} + w_0)$ where $\text{sgn}(z) = \begin{cases} -1 & \text{if } z < 0 \\ +1 & \text{if } z \geq 0 \end{cases}$

Soft classifier: $f(\mathbf{x}) = h(\mathbf{w}^T \mathbf{x} + w_0)$ where $h(z) = \begin{cases} -1 & \text{if } z < -1 \\ z & \text{if } -1 \leq z \leq 1 \\ +1 & \text{if } z > 1 \end{cases}$

# Linear SVMs: Non-separable Case

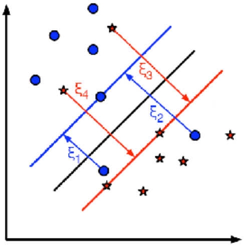


An example of two non-separable classes.

# Linear SVMs: Non-separable Case

**Three categories of input samples:**

- Samples fall outside the band and are correctly classified:
$$y_i(\mathbf{w}^T\mathbf{x}_i + w_0) \geq 1$$
- Samples fall inside the band and are correctly classified:
$$0 \leq y_i(\mathbf{w}^T\mathbf{x}_i + w_0) < 1$$
- Samples are misclassified:
$$y_i(\mathbf{w}^T\mathbf{x}_i + w_0) < 0$$



All 3 categories can be described as:
$$y_i(\mathbf{w}^T\mathbf{x}_i + w_0) \geq 1 - \xi_i$$

where $\xi_i$ is known as *slack variable*. First category: $\xi_i = 0$; Second category: $0 < \xi_i \leq 1$; Third case: $\xi_i > 1$.

# Linear SVMs: Non-separable Case

We want to maximise the margin and minimise the number of misclassified point (minimise the margin violations). We formulate the constrained optimisation problem as:

$$\boxed{\begin{gathered} \min_{\mathbf{w},w_0,\xi} \quad J(\mathbf{w},\xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + w_0) \geq 1 - \xi_i, \quad i = 1,2,\ldots,N \\ \xi_i \geq 0, \quad i = 1,2,\ldots,N \end{gathered}}$$

where $\xi = \begin{bmatrix} \xi_1 & \xi_2 & \ldots & \xi_N \end{bmatrix}$ and $0 \leq C \leq +\infty$ is a pre-set constant scalar, which controls the influence of the two competing terms.

This is know as the *soft-margin method*, the classifier is know as soft-margin classifier (do not confuse with soft/hard classifiers).

# Linear SVMs: Non-separable Case

The constrained optimisation problem is formulated as the following primal problem using the method of Lagrange multipliers:

$$\text{Primal problem: } \mathscr{L}(\mathbf{w}, w_0, \xi, \lambda, \mu) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\mu_i\xi_i - \sum_{i=1}^{N}\lambda_i(y_i(\mathbf{w}^T\mathbf{x}_i + w_0) - 1 + \xi_i))$$

where $\mu = \left[\begin{array}{cccc} \mu_1 & \mu_2 & \dots & \mu_N \end{array}\right]$ and $\lambda = \left[\begin{array}{cccc} \lambda_1 & \lambda_2 & \dots & \lambda_N \end{array}\right]$ are Lagrange multipliers.

The above primal problem can be transformed to the following dual problem:

$$\text{Dual problem: } \min_{\mathbf{w}, w_0, \xi} \max_{\lambda \geq 0, \mu \geq 0} \mathscr{L}(\mathbf{w}, w_0, \xi, \lambda, \mu)$$

# Linear SVMs: Non-separable Case

$$\frac{\partial \mathscr{L}(\mathbf{w}, w_0, \xi, \lambda, \mu)}{\partial \mathbf{w}} = \mathbf{0} \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{N} \lambda_i y_i \mathbf{x}_i \tag{4}$$

$$\frac{\partial \mathscr{L}(\mathbf{w}, w_0, \xi, \lambda, \mu)}{\partial w_0} = 0 \quad \Rightarrow \quad \sum_{i=1}^{N} \lambda_i y_i = 0 \tag{5}$$

$$\frac{\partial \mathscr{L}(\mathbf{w}, w_0, \xi, \lambda, \mu)}{\partial \xi_i} = 0 \quad \Rightarrow \quad C - \mu_i - \lambda_i = 0 \tag{6}$$

Putting (4), (5) and (6) into $\mathscr{L}(\mathbf{w}, w_0, \xi, \lambda, \mu)$, we have

$$\begin{aligned}
\mathscr{L}(\lambda, \xi, \mu) &= \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \lambda_i (y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i) \\
&= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \lambda_i y_i \mathbf{w}^T \mathbf{x}_i - \sum_{i=1}^{N} \lambda_i y_i w_0 + \sum_{i=1}^{N} \lambda_i - \sum_{i=1}^{N} \lambda_i \xi_i \\
&= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^{N} \lambda_i y_i w_0 + \sum_{i=1}^{N} \lambda_i - \sum_{i=1}^{N} (C - \mu_i)\xi_i \\
&= \sum^{N} \lambda_i - \frac{1}{} \sum^{N}\sum^{N} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum_{i=1}^{N} \mu_i \xi_i .
\end{aligned}$$

# Linear SVMs: Non-separable Case

By minimising $\mathscr{L}(\lambda,\xi,\mu)$ with respective to $\xi$, $\mu_i\xi_i = 0$ has to be achieved. $\mathscr{L}(\lambda,\xi,\mu)$ is reduced to:

$$\mathscr{L}(\lambda) = \sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j.$$

The dual problem is then reduced to:

$$\max_{\lambda\geq 0}\left(\sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j\right) \tag{7}$$

$$\text{subject to } \sum_{i=1}^{N}\lambda_i y_i = 0,$$

$$0 \leq \lambda_i \leq C, i = 1,2,\ldots,N$$

# Linear SVMs: Non-separable Case

- The same remarks from linearly separable case apply.
- This dual problem is the same as that of linearly separable case except a bound is given to $\lambda_i$.
- Some input samples corresponding to $\xi_i \neq 0$ (within the margin or being misclassified), leading to $\mu_i$ and $\lambda_i = C$, have largest contribution to the final solution $\mathbf{w}$.

# Linear SVMs: Non-separable Case

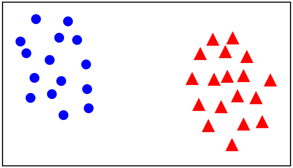
**Summary:** The linear SVM classifier (non-separable case) can be found by solving the solution ($\mathbf{w}$, $w_0$, $\xi_i$, $\lambda_i$, $\mu_i$) to the following conditions:

$$\boxed{\begin{aligned}
\frac{\partial \mathscr{L}(\mathbf{w}, w_0, \lambda, \xi, \mu)}{\partial \mathbf{w}} &= \mathbf{0} \\
\frac{\partial \mathscr{L}(\mathbf{w}, w_0, \lambda, \xi, \mu)}{\partial w_0} &= 0 \\
\frac{\partial \mathscr{L}(\mathbf{w}, w_0, \lambda, \xi, \mu)}{\partial \xi_i} &= 0 \\
\mu_i \geq 0, \quad \lambda_i &\geq 0, \quad i = 1, 2, \ldots, N \\
\mu_i \xi_i &= 0, \quad i = 1, 2, \ldots, N \\
\lambda_i \big( y_i (\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i \big) &= 0, \quad i = 1, 2, \ldots, N
\end{aligned}}$$

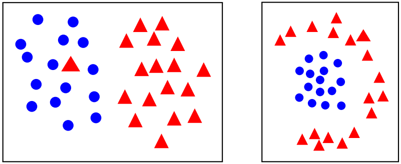
Hard classifier: $f(\mathbf{x}) = \text{sgn}(\mathbf{w}^T\mathbf{x} + w_0)$, Soft classifier: $f(\mathbf{x}) = h(\mathbf{w}^T\mathbf{x} + w_0)$

# Nonlinear SVMs

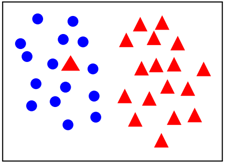
**Linearly separable case:**



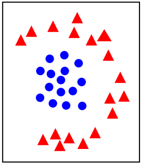
**Nonlinearly separable case:**



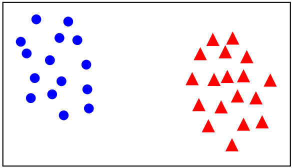
Does it exist a mapping $\Phi(\cdot)$ so as:



or



$\Rightarrow \Phi(\cdot) \Rightarrow$



# Nonlinear SVMs

- Feature mapping: $\mathbf{z}_i = \Phi(\mathbf{x}_i)$, $i$ = 1, 2, . . ., $N$.
- In the above analysis for linear SVMs, instead of using $\mathbf{x}_i$, we use $\mathbf{z}_i$.
- The same analysis results (formulas) can be applied.
- Linear SVMs: $\mathbf{w} = \sum_{i \in SVs} \lambda_i y_i \mathbf{x}_i \Rightarrow$ hyperplane: $\mathbf{w}^T \mathbf{x} + w_0 = 0$
- Nonlinear SVMs: $\mathbf{w} = \sum_{i \in SVs} \lambda_i y_i \mathbf{z}_i \Rightarrow$ hyperplane: $\mathbf{w}\mathbf{z} + w_0 = 0$

$$\text{Hyperplance: } \sum_{i \in SVs} \lambda_i y_i \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}) + w_0 = 0$$

- Hard classifier: $f(\mathbf{x}) = \text{sgn}(\sum_{i \in SVs} \lambda_i y_i \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}) + w_0)$
- Soft classifier: $f(\mathbf{x}) = h(\sum_{i \in SVs} \lambda_i y_i \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}) + w_0)$

# Nonlinear SVMs

- Certain Kernels that satisfy *Mercer's Theorem* allow mapping to high-dimensional feature space implicitly:

$$K(\mathbf{x}_i, \mathbf{x}) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x})$$

- Instead of find the mapping function $\Phi(\cdot)$, it is easier to find the kernel function $K(\cdot)$.
- Hard classifier: $f(\mathbf{x}) = \text{sgn}(\sum_{i \in SVs} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) + w_0)$
- Soft classifier: $f(\mathbf{x}) = h(\sum_{i \in SVs} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) + w_0)$

# Nonlinear SVMs

**Commonly used kernels:**

- Linear kernel (standard inner product): $K(\mathbf{x}_i, \mathbf{x}) = \mathbf{x}_i^T \mathbf{x} + c$, $c$ is an optional constant.
- Polynomial kernel of degree $q$: $K(\mathbf{x}_i, \mathbf{x}) = (\alpha \mathbf{x}_i^T \mathbf{x} + c)^q, \quad q > 0$
- Radial basis function (RBF) kernel (exponential kernel): $K(\mathbf{x}_i, \mathbf{x}) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{2\sigma^2}}$
- Multi-quadratic kernel: $K(\mathbf{x}_i, \mathbf{x}) = \sqrt{\|\mathbf{x}_i - \mathbf{x}\|^2 + c}$
- Inverse multi-quadratic kernel: $K(\mathbf{x}_i, \mathbf{x}) = \frac{1}{\sqrt{\|\mathbf{x}_i - \mathbf{x}\|^2 + c}}$
- Power kernel: $K(\mathbf{x}_i, \mathbf{x}) = -\|\mathbf{x}_i - \mathbf{x}\|^q$
- Log kernel: $K(\mathbf{x}_i, \mathbf{x}) = -\log(\|\mathbf{x}_i - \mathbf{x}\|^q + 1)$
- Sigmoid Function (Hyperbolic Tangent): $K(\mathbf{x}_i, \mathbf{x}) = \tanh(\beta \mathbf{x}_i^T \mathbf{x} + \gamma)$
- A kernel can be constructed from other kernels.
  - A linear combination of kernels: $\sum_k \alpha_k K_k(\cdot)$, $\alpha_k > 0, \forall k$
  - Product of kernels: $\prod_k \alpha K_k(\cdot)$ (so as $K(\cdot)^q$)

# Nonlinear SVMs

**Why kernel functions?**

Example: Consider the quadratic kernel (polynomial kernel of degree two, without offset term): $K(\mathbf{z},\mathbf{x}) = (\mathbf{z}^T\mathbf{x})^2 = z_1^2x_1^2 + 2z_1z_2x_1x_2 + z_2^2x_2^2$.

It can be shown that the feature mapping function is:

$$\Phi(\mathbf{x}) = \begin{bmatrix} x_1 \\ \sqrt{2}x_1x_2 \\ x_2 \end{bmatrix} \Rightarrow K(\mathbf{z},\mathbf{x}) = \Phi(\mathbf{z})^T\Phi(\mathbf{x}) = (\mathbf{z}^T\mathbf{x})^2$$

- The original feature space is mapped to a higher-dimensional feature space through a feature mapping function $\Phi(\cdot)$.
- Instead of computing $\Phi(\cdot)$ and then $\Phi(\cdot)^T\Phi(\cdot)$ (two-step computation), computing the kernel function $K(\cdot)$ can be done in one step saving the computational demand especially for feature mapping function of higher dimensional space.

# Nonlinear SVMs

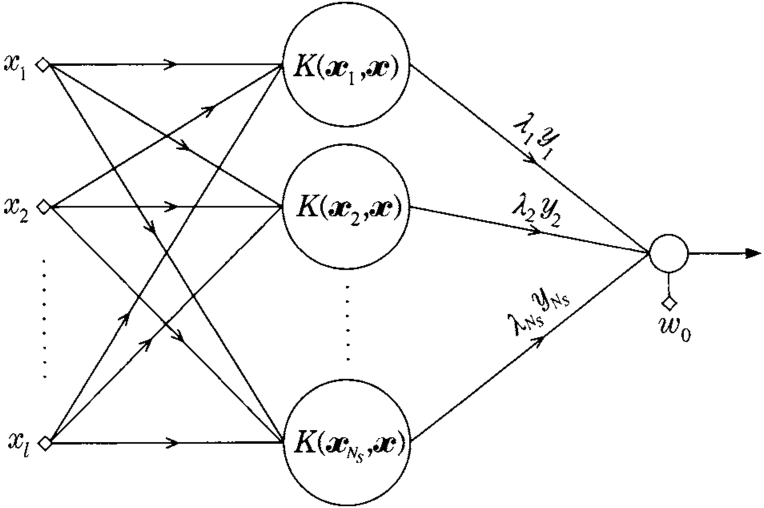

$x_1$
$x_2$
$x_l$
$K(\boldsymbol{x}_1,\boldsymbol{x})$
$K(\boldsymbol{x}_2,\boldsymbol{x})$
$K(\boldsymbol{x}_{N_S},\boldsymbol{x})$
$\lambda_1 y_1$
$\lambda_2 y_2$
$\lambda_{Ns} y_{Ns}$
$w_0$

Figure 4: A diagram of SVM classifier with kernel functions.

# Nonlinear SVMs

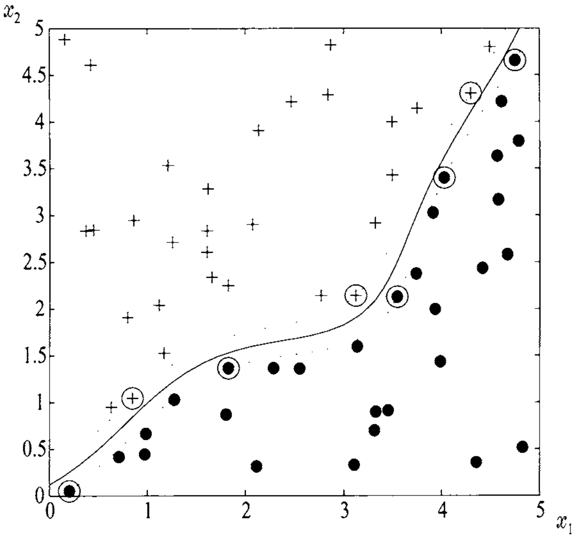


Figure 5: A nonlinearly separable classification example using nonlinear SVM classifier.

# Nonlinear SVMs

- When RBF kernel function is used, the SVM architecture is the same as the RBF network structure. However, the number of hidden units and the centres are determined by the optimisation procedure.

- When sigmoid kernel function is used, the SVM architecture is the same as a two-layer fully-connected feed-forward neural network structure. However, the number of hidden units is determined by the optimisation procedure.

- There is no systematic method to determine the best Kernel function and its parameters, and the parameter $C$ (hyper-parameters), which are usually chosen by trial and error.

# Applications

- Speaker verification
- Face detection
- Hand-writing recognition
- Biomedical
    - Cancer diagnosis
    - Epilepsy Diagnosis (EEG)
    - Cardiac Arrhythmia (ECG)
    - Cardiovascular Disease
- Many more

# Multi-class SVMs

- SVM classifiers is a *binary classifier*, which can handle only two-class problems.
- Multi-class SVM classifiers can be built by combining two-class SVM classifiers.

**Multi-class classification problems:**

Given a dataset: $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ and each data point $\mathbf{x}_i$ belongs to class $C_i \in \{1, 2, \ldots, R\}$; $i$ = 1, 2, ..., $N$, design a classifier which can tell which class the data $\mathbf{x}_i$ belongs to.

# Multi-class SVMs

**Approaches combining SVMs**

- One against one
- One against all
- Binary decision tree
- Binary coded

These approaches combining a number of two-class SVMs (linear or nonlinear) for multi-class classification.

# Multi-class SVMs

**One-against-one approach:**

- Number of classifiers: $\frac{R(R-1)}{2}$ 2-class classifiers are required for $R$-class problem.
- Learning: A 2-class SVM classifiers is trained for each pair of classes, i.e., $C_i$ versus $C_j$. For example, the 2-class hard SVM classifier sgn$\left(\mathbf{w}_{ij}^{(ij)}\mathbf{x} + w_0^{(ij)}\right)$ is able to classify if the input $\mathbf{x}$ belongs to class $C_i$ (using label $+1$) or $C_j$ (using label $-1$).
- Decision: Choose class with majority votes.
- For example, considering 3 classes, we need to have $\frac{3(3-1)}{2} = 3$ classifiers, i.e., 1 against 2, 2 against 3 and 3 against 1. If the output of 1-against-2 classifier is Class 1; 2-against-3 classifier is Class 2/3; 3-against-1 classifier is Class 1, the major vote is Class 1, which is the final decision.
- Some regions cannot be classified.

# Multi-class SVMs

**One-against-one approach:**

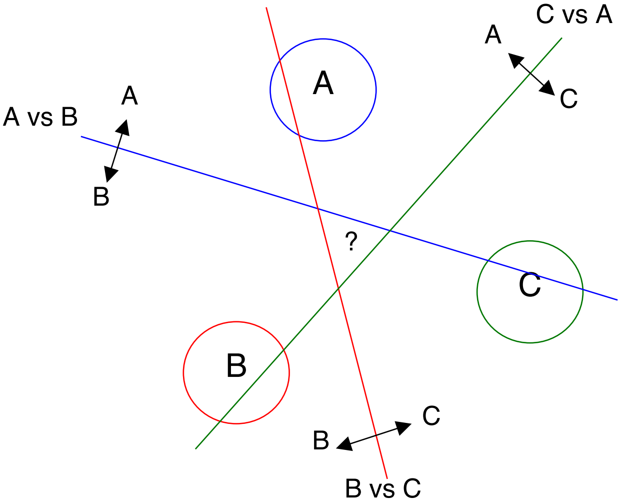


Figure 6: One-against-one approach for 3 classes.

# Multi-class SVMs

**One-against-all approach (soft SVM classifiers):**

- Number of classifiers: $R$ 2-class classifiers are required for $R$-class problem.
- Learning: A 2-class SVM classifiers is trained for $C_R$ versus all $C_j, j \neq R$. For example, the 2-class hard SVM classifier $\mathbf{w}_i^{(i)}\mathbf{x} + w_0^{(i)}$ is able to classify if the input $\mathbf{x}$ belongs to class $C_i$ (positive value) or the rest (negative value).
- Decision: Choose class with largest value.
- For example, considering 3 classes, we need to have $R = 3$ classifiers, i.e., 1 against 2&3, 2 against 1&3, and 3 against 1&2. If the output of 1-against-2&3 classifier is 1.6; 2-against-1&3 classifier is 0.2; 3-against-1&2 classifier is $-0.8$, the largest value is 1.6, the final decision is Class 1.
- Some regions cannot be classified using hard SVM classifiers.
- All regions can be classified using soft SVM classifiers. However, misclassification may happen in the region near the boundary.

# Multi-class SVMs using hard SVM classifiers

**One-against-all approach:**

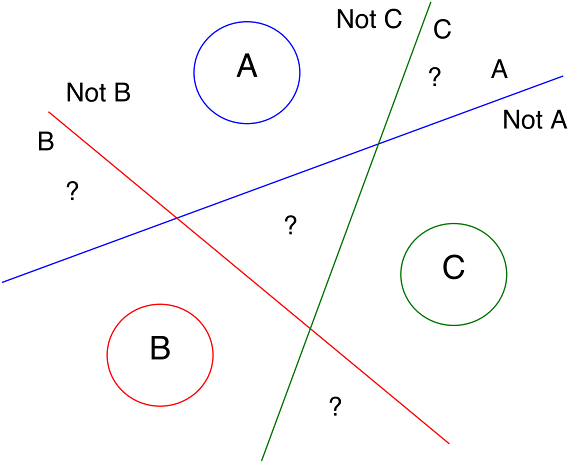


Figure 7: One-against-all approach for 3 classes using hard SVM classifiers.

# Multi-class SVMs

**One-against-all approach with soft SVM classifiers:**

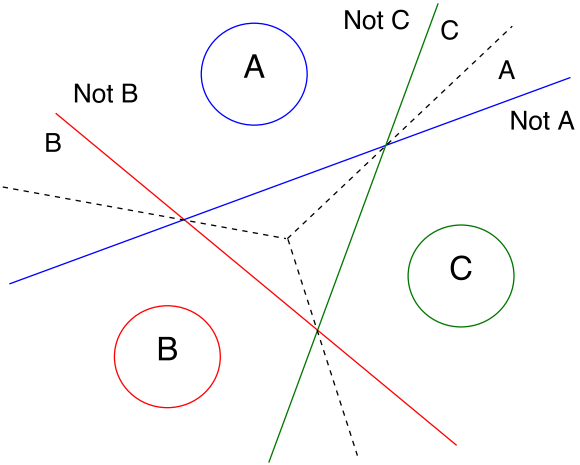


Figure 8: One-against-all approach for 3 classes using soft SVM classifiers.

# Multi-class SVMs

**Binary decision tree:** It requires $R-1$ SVMs for a problem with $R$ classes. It consults at most $\lceil \log_2 R \rceil$ SVMs to make a final decision for a sample $\mathbf{x}$. The classification accuracy depends heavily on the SVMs in the upper levels.

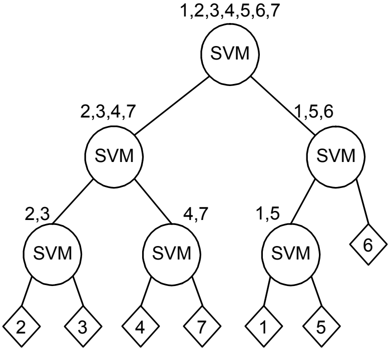


Figure 9: Binary decision tree approach for 7 classes.

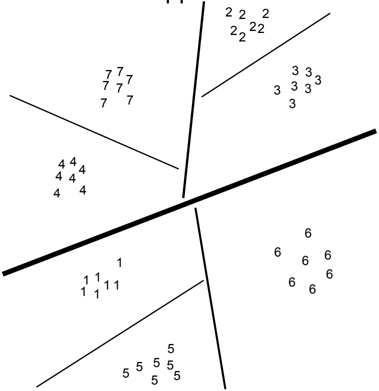


Figure 10: Binary decision tree divisions for 7 classes.

# Multi-class SVMs

**Binary coded approach:** It requires $\lceil \log_2 R \rceil$ SVMs for a problem with $R$ classes.

Example: Considering a classification of 7 classes, it requires $\lceil \log_2 7 \rceil = \lceil 2.8074 \rceil = 3$ SVMs.

| Class | SVM 1 | SVM 2 | SVM 3 |
|---|---|---|---|
| 1 | $+1$ | $+1$ | $+1$ |
| 2 | $+1$ | $+1$ | $-1$ |
| 3 | $+1$ | $-1$ | $+1$ |
| 4 | $+1$ | $-1$ | $-1$ |
| 5 | $-1$ | $+1$ | $+1$ |
| 6 | $-1$ | $+1$ | $-1$ |
| 7 | $-1$ | $-1$ | $+1$ |

**Class assignments:**

- SVM 1: $\underbrace{1234}_{+1} \mid \underbrace{567}_{-1}$
- SVM 2: 1256|347
- SVM 3: 1357|246

# Conclusion

- Linear SVM classifiers with hard/soft margin are designed through optimisation procedure.
- Nonlinear SVM classifiers using kernel functions has been introduced to deal with non-separable classification problems.
- Support Vector Machines are versatile and effective tools for classification problems.
- Multi-class classification problem can be dealt with by combining binary-class SVM classifiers.